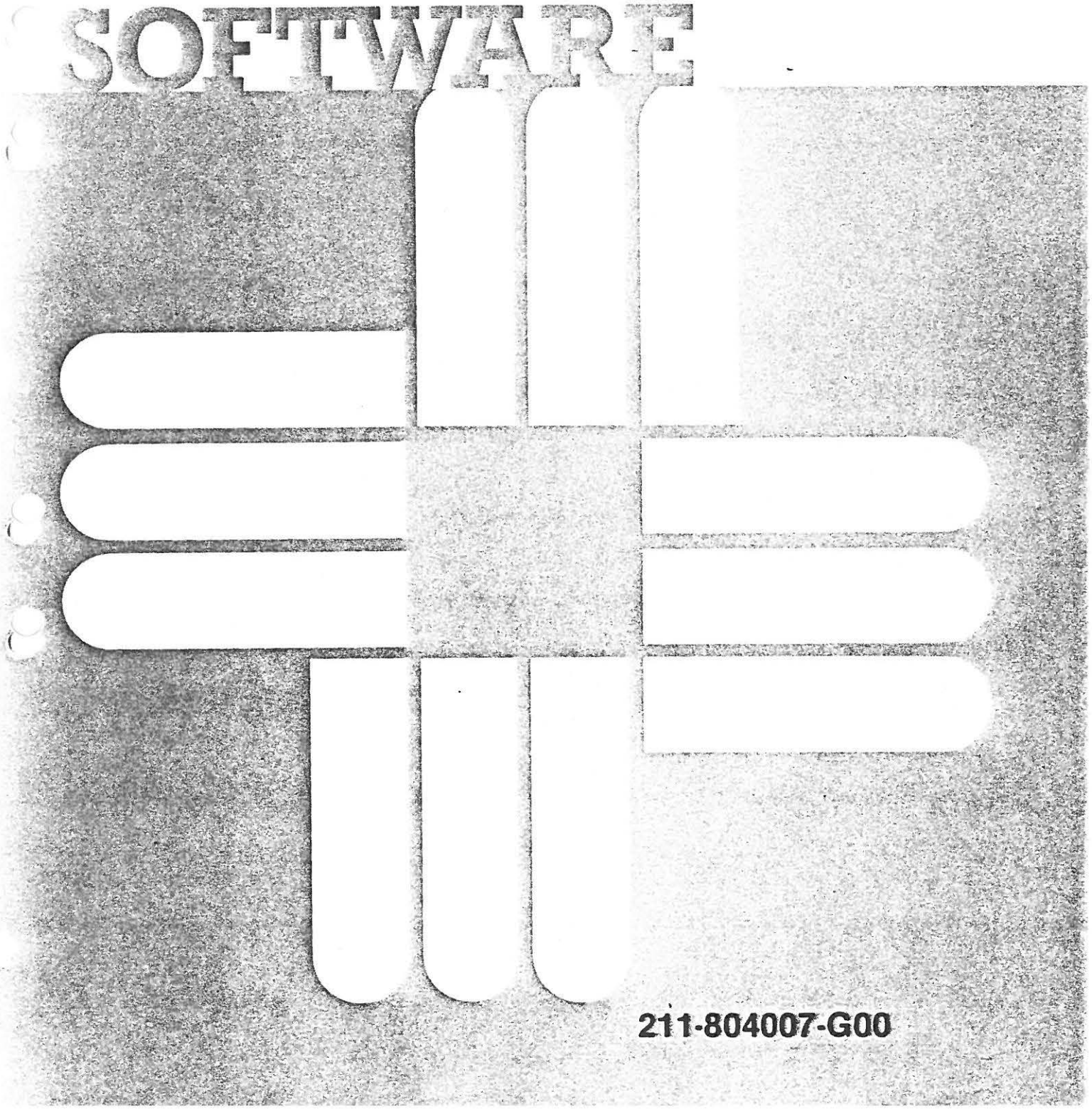
MODCOMP
MODULAR COMPUTER SYSTEMS, INC.

MAX IV/MAX 32 Programmer's Reference Manual

Assembler Cross-Reference (XREF) Models 8240 and 8250



211-804007-G00

MANUAL HISTORY

Manual Title: MAX IV/MAX 32 PROGRAMMER'S REFERENCE MANUAL, ASSEMBLER CROSS-REFERENCE (XREF)

Order Number: 211-804007-G00

Model Numbers: 8240 and 8250

Software Part Number: 611599-120 G.0

Revision Level	Date Issued	Description
A00	07/76	Initial Issue (A.0).
C00	05/77	Reissue (C.0).
C01	02/78	Reissue (C.1).
E00	06/79	Reissue (E.0).
F00	02/82	Reissue (F.0).
G00	05/85	Reissue (G.0). Included references to 32-bit Assembly language source programs as valid input to XREF. Manual format was also reorganized.

Contents subject to change without notice.

Copyright © 1985, by Modular Computer Systems, Inc.
All Rights Reserved.
Printed in the United States of America.

PROPRIETARY INFORMATION

THIS MANUAL CONTAINS CONFIDENTIAL PROPRIETARY INFORMATION PROTECTED BY SOFTWARE PROPERTY RIGHTS AND IS MADE AVAILABLE UPON THE CONDITION THAT THE SOFTWARE CONTAINED HEREIN WILL BE HELD IN ABSOLUTE CONFIDENCE AND MAY NOT BE DISCLOSED IN WHOLE OR IN PART TO OTHERS WITHOUT THE PRIOR WRITTEN PERMISSION OF MODULAR COMPUTER SYSTEMS, INC

FOR GOVERNMENT USE THE FOLLOWING SHALL APPLY

RESTRICTED RIGHTS LEGEND

USE, DUPLICATION OR DISCLOSURE BY THE GOVERNMENT IS SUBJECT TO RESTRICTIONS AS SET FORTH IN PARAGRAPH (b) (3) (B) OF THE RIGHTS IN TECHNICAL DATA AND COMPUTER SOFTWARE CLAUSE IN DAR7-104.9(a) (OR SUCH EQUIVALENT FPR, NASPR, AND/OR DOEPR CLAUSE AS MAY BE APPLICABLE).

MODULAR COMPUTER SYSTEMS, INC.
1650 WEST McNAB ROAD
FORT LAUDERDALE, FL 33309

PREFACE

Audience
This manual is directed to system programmers who have written programs in MODCOMP Assembly language and want to cross-reference the symbols used in those programs. Some familiarity with the MODOCMP Source Editor (SED) or Screen Editor (MSED) and the MAX IV or MAX 32 operating system is also assumed.

Subject
This manual describes XREF, a program used to generate cross-reference listings of symbols used in MODCOMP Assembly language source programs. It documents the resources needed to run XREF, its commands and options, and the format of the cross-reference output listings.

Product Requirements
XREF runs under a MAX IV or MAX 32 operating system.

Product Support
The Assembler source files used as input to XREF must be maintained by the Source Editor (SED) or Screen Editor (MSED). In some cases, Assembly programs must be assembled with either the M5A or ASM32 Assembler before they can be processed by XREF.

Related Publications
Refer to the following manuals for additional information. When ordering manuals, use the manual order number listed below. The most current revision level (REV) will be shipped.

Manual Order Number	Manual Title
210-804001-REV	MODCOMP ASSEMBLERS MAX IV/MAX 32 Language Reference Manual (LRM)
211-804005-REV	SOURCE EDITOR MAX IV Programmer's Reference Manual (PRM)
211-804002-REV	NONRESIDENT JOB CONTROL AND BATCH FACILITIES MAX IV/MAX 32 Programmer's Reference Manual (PRM)

Special Symbols and Notations
A revision bar (|) located in the margin of a page indicates a change to the manual. This change generally represents a technical change to the product due to product revision. Revision bars are also used in the Table of Contents to flag the general location of changes in the text.

MODCOMP Product Training
MODCOMP's Education Services department provides training courses on many hardware and software products at our Training Center in Fort Lauderdale, Florida. On-site courses at customer locations can also be arranged. For more information about the courses offered by Education Services, contact the MODCOMP Training Registrar.

TABLE OF CONTENTS

Page

CHAPTER 1 INTRODUCTION

1.1 OVERVIEW OF XREF 1-1
1.2 SYMBOLS 1-1
1.3 SYSTEM REQUIREMENTS 1-2
1.4 SUMMARY OF COMMANDS 1-2
1.5 SYSTEM FILES 1-3
1.6 OPTIONS 1-3
1.7 ACCESSING XREF 1-4
1.8 LIMITATIONS 1-5

CHAPTER 2 XREF COMMANDS

2.1 COMMAND INPUT 2-1
2.2 SYMBOL SUPPRESSION 2-1
2.3 COMMAND DESCRIPTIONS 2-2
:BIA 2-3
:CON 3-4
:DEO 2-5
:EXI 2-6
:NEJ 2-7
:NOE 2-8
:NUM 2-9
:OPC 2-10
:STA 2-11
:STO 2-12
:WRK 2-13

CHAPTER 3 XREF OUTPUT

3.1 OUTPUT FORMAT 3-1

3.1.1 Type One Indicators 3-1
3.1.2 Type Two Indicators 3-2

3.2 DEVICE LIMITATIONS FOR PRINTED OUTPUT 3-5
3.3 ERROR MESSAGES 3-5

INDEX I-1

LIST OF FIGURES

3-1 Assembler Source Input to XREF 3-3
3-2 XREF Output Listing 3-4

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF XREF

The Assembler Cross-Reference program (XREF) is used to generate cross-reference listings of symbols used in Assembly language source programs. Cross-reference lists are useful during the process of debugging an Assembler program in that values of symbols and references to them are easily identifiable.

Input to XREF is Assembly language source code written for either the MODCOMP 16-bit Assembler (M5A) or 32-bit Assembler (ASM32). The source files must have been created using either the MODCOMP Source Editor (SED) or Screen Editor (MSED), and can be in ASCII compressed or uncompressed format. Depending on the setting of various options and the use of certain commands, Assembler source input can be in SED random or sequential format.

XREF examines only the first 72 bytes of each source record (after uncompression, if necessary). An ASCII 'null' character can be used to force XREF to terminate reading of a record before byte 72. XREF ignores Assembler comment lines (which are indicated by a asterisk in column one), text strings within double quote delimiters, lines that specify a subtitle (using the STL directive), and all lines that specify a title (using the TTL directive) except the first. The first TTL directive is used as a header for the output listing.

XREF provides a number of options and commands to tailor cross-reference listings to suit the requirements of particular programming tasks. Options are discussed in Section 1.6. XREF commands are summarized in Section 1.4 and explained in more detail in Chapter 2.

1.2 SYMBOLS

Detailed information about the use of symbols in Assembly language programming is in the MODCOMP Assemblers, MAX IV/MAX 32 Language Reference Manual. To summarize, a symbol is any symbolic name constructed according to the following rules:

- The first character must be alphabetic.
- All characters after the first must be in the Compressed Alphanumeric (CAN) set. CAN-code characters include the letters A - Z (uppercase only), digits 0 - 9, the dollar sign ($), period (.), and colon (:).
- The name can be any length but only the first six characters are significant and must be unique. Any characters after the first six are ignored.
- A blank space or any character not in the CAN-code set terminates the symbol.

XREF cross-references symbols that occur:

- In the label or argument field of an Assembler statement.
- In the operation field, other than the first subfield, unless the symbol was previously used in the label field (that is, macro calls but not operation codes or compiler directives).

XREF commands or options can be used to include symbols that appear in the first subfield of the operation field and other nonsymbolic strings in the cross-reference. A symbol suppression feature can also be used to eliminate specific symbols from the cross-reference.

1.3 SYSTEM REQUIREMENTS

XREF can be used effectively on CLASSIC Series systems that have the required peripherals to support the files that XREF uses. If the Assembler programs used as input to XREF contain INS directives to read in other files, the system should have a rewindable scratch device. If a rewindable scratch device is not available, the cross-reference may not contain all references to symbols in the inserted file(s).

To ensure that all symbols in insert files are included in the cross-reference, first prepare a source file that contains all of the statements to be processed. The simplest way to do this is to execute the original program and the insert files through pass 1 of the Assembler. The SC option must be set and logical file SC must be assigned to a system scratch device. At the end of pass 1 processing through the Assembler, a file mark must be written at the end of the file, and the file must be rewound and used as source input to XREF.

For practical purposes you may be forced to accept the ordering of inserts and original source produced by the Assembler. However, cross-reference listings may be more useful if they do not include information from insert files, but represent only the original source program.

1.4 SUMMARY OF COMMANDS

The following is a summary of XREF commands. Each command is explained in more detail in Chapter 2. Note that some commands and options perform exactly the same function. When this is the case, either one can be used. Section 1.6 contains a list of all of the options relevant to XREF.

Command	Option	Description
:BIA n		Sets a line number bias on the listed output.
:CON	$CN	Continues processing even if symbol table becomes full.
:DEO	$DE	Terminates cross-referencing when a double end-of-file mark is encountered.
:EXI		Exits XREF and returns to Job Control.
:NEJ	$NJ	Suppresses page ejects on listed output.
:NOE	$NE	Terminates cross-referencing when an end-of-file mark is encountered instead of when an END directive is encountered.
non-CAN-code character or a blank space		Terminates XREF command input and generates the cross-reference listing.

Command	Option	Description
:NUM	$NM	Includes numerical and CAN-code constants as symbols in the cross-reference.
:OPC	$OP	Includes operation codes as symbols in the cross-reference.
:STA n		Starts cross-referencing at source line n.
:STO n		Stops cross-referencing at souce line n.
symbol string		Suppresses the cross-referencing of the specified symbol(s).
:WRK	$WK	Source input is in SED random format from logical file WRK.

1.5 SYSTEM FILES

XREF uses the following logical files during processing:

- AI Commands are read from this file if the AO option is set when XREF is executed.
- CI Commands are read from this file if the AO option is not set when XREF is executed.
- LO The cross-reference output listing is written to this file.
- SC Assembler source input is read from this file if the SC option is set when XREF is executed.
- SI Assembler source input is read from this file if the SC option is not set when XREF is executed.
- WRK Assembler source input is read from the workfile if the $WK option or :WRK command is in effect. The source file must be in SED random format.

1.6 OPTIONS

System and user options are used to vary the type of cross-reference listing produced by XREF. As indicated in the summary of commands (Section 1.4), some options and commands perform the same function, so either can be used. Options are specified through Job Control on the $EXECUTE directive when XREF is invoked. User options are identified by a leading dollar sign.

Option

- LO Generates a sequentially numbered listing of the Assembler source program in addition to the cross-reference listing. The listing of the Assembler source file is the same as the one that is generated using the LIST command in SED. For XREF processing, LO is the default.
- NOLO Suppresses the Assembler source file listing.

Option	
SC	Reads Assembler source input from logical file SC. If this option is specified, XREF alters the output line numbers if any insert files were included in the original Assembler souce program. For XREF processing, SC is the default.
NOSC	Reads Assembler source input from logical file SI.
AO	Reads XREF commands from logical file AI. If this option is not specified, XREF reads commands from logical file CI.
$NC	XREF neither expects nor reads commands, and proceeds to read Assembler source input from logical file SI, SC, or WRK.
$CN	Clears the symbol table and continues to generate a cross-reference listing even after the symbol table becomes full.
$DE	Reads Assembler source code until a double end-of-file mark is encountered.
$NJ	Prints a double line feed instead of a page eject, except when a bottom-of-form is encountered.
$NE	Ignores Assembler END directives and continues reading source files until an end-of-file mark is encountered. This option is useful for cross-referencing several programs one after the other or for ignoring END directives within macro files.
$NM	Includes numeric and CAN-code constants as symbols in the cross-reference listing, but does not print the at sign (@) that precedes CAN-code constants or the pound sign (#) that precedes hexadecimal constants.
$OP	Includes all operation codes (opcodes) in the cross-reference listing. This option is useful for cross- referencing macro files.
$WK	Reads Assembler source input from logical file WRK, which must be a valid random file. If this option is specified, XREF ignores the SC and $DE options and any :DEO commands. The random file must be maintained under SED or MSED.

System options (those without a leading dollar sign) and user options are both specified on the $EXECUTE directive. For example:

```
$EXE XREF,,NOLO,NOSC,$NE,$OP
```

or

```
$EXE XREF,,LO,AO,$NM,$CN
```

1.7 ACCESSING XREF

Before attempting to run XREF, the system GO/NOGO option must be set to GO. Otherwise, Job Control will not allow XREF to execute.

The following job stream produces a standard cross-reference listing of an Assembler source file from a User Source Library (USL). Output is directed to a line printer. The $EXECUTE command line shown in the following example is the minimum requirement to execute XREF:

```
$JOB
$ASS SI USL
$ASS LO LP
$POS name-of-Assembler-file-to-be-cross-referenced
$EXE XREF,,$NC,NOSC
$JOB
```

To direct output to your terminal, omit the assignment of LO to LP.

By default, the cross-reference terminates when XREF encounters an END directive within the Assembler source file since no options or commands were used to specify otherwise.

XREF should be cataloged on your system with ALLOCATE ALL in effect. This ensures that XREF has enough space to generate symbol tables during applications. If XREF was not cataloged with ALLOCATE ALL, add the following statement to the job stream prior to the $EXECUTE XREF statement:

```
$ALLOCATE ALL
```

A procedure can be set up to execute an Assembler program and then cross-reference the symbols with XREF. The job stream required to do this is listed below. In this example, five arguments can be passed to the procedure. %1 is the name of the source file to be used as input to the Assembler and XREF. %2 and %3 are options that can be passed to the Assembler. %4 and %5 are options that can be passed to XREF.

This example assumes that no XREF commands will be entered, since XREF is invoked with the $NC option. It also assumes that the Assembly file used as input is located on logical file SI, which is assigned to your USL. To assemble a program with the 32-bit Assembler instead of M5A, substitute ASM32 on the $EXECUTE directive line.

```
$PROC name-of-procedure %1,%2,%3,%4,%5
$ASSIGN SI USL
$POS %1
$EXE M5A,,%2,%3
$POS %1
$EXE XREF,,%4,%5,$NC
```

1.8 LIMITATIONS

The syntax recognized by XREF does not correspond exactly to the syntax of the M5A or ASM32 Assemblers. Although the differences are slight, the semantic rules incorporated into XREF are more elementary than the semantic rules of the Assemblers. Therefore, XREF may occasionally classify some strings as symbols that the Assemblers would not.

XREF does not cross-reference symbols that appear in the operation code field of a source statement unless that symbol has already appeared as a label elsewhere in the program or unless the :OPC command is used. Thus, if labels that match operation codes are used within a source program, references to those symbols will appear on the cross-reference listing.

Any strings that are not Assembler symbols but are classified as symbols by XREF can be suppressed from the cross-reference. Refer to Section 2.2 for more information on this symbol suppression feature.

If the symbol table becomes full when XREF is generating the cross-reference, the program will abort with an OST code. To override this, use the :CON command. If :CON is in effect and the symbol table reaches capacity, XREF prints the current cross-reference, clears the entries from its symbol table, and continues processing the current program.

CHAPTER 2
XREF COMMANDS

2.1 COMMAND INPUT

XREF commands can be issued in two ways: either on subsequent lines after the $EXECUTE directive (that is, from the CI file) or, if the AO option is set, commands can be read from a file assigned to logical file AI.

To terminate command input to XREF and begin generating the cross-reference, enter any character that is not alphabetic and not a colon in column one on a line by itself. To end command input to XREF without generating a cross-reference, use the :EXI command.

If the $NC option is not in effect, XREF reads commands from either logical file CI or AI. If the first character read from either of these logical files is a colon, XREF command input is accepted. If the first character is alphabetic, this indicates that you are specifying strings which should not be considered for cross-referencing (refer to the following section for more information). If the first character is anything else, command input to XREF is terminated and the cross-reference is generated.

2.2 SYMBOL SUPPRESSION

In addition to the commands listed in the remainder of this chapter, XREF provides a symbol suppression feature. To prevent certain symbols in the Assembler source program from being cross-referenced, list those symbol names in the XREF command stream. XREF assumes that any line that begins with an alphabetical character is a request for symbol suppression. Any number of symbols can be listed, and each one must be separated by a comma. The list can be continued onto the next line; in this case, do not specify a comma after the last symbol name on the line if the list is to be continued on the next line. If the line to be continued has a comma as the last character, XREF ends command input and begins generating the cross-reference. The following command stream illustrates the use of the symbol suppression feature. Refer to the remainder of this chapter for information on the commands used in this example.

```
$POS name-of-Assembler-source-file-to-be-cross-referenced
$EXE XREF,,NOLO,NOSC
:STA 100
:STO 1000
LAB1,R1,R2,R3,R4
R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15
ONE,TWO
*
```

This example starts cross-referencing at line 100 and and stops after processing line 1000 of the Assembler source file. XREF expects to find the source file on logical file SI since the NOSC option was specified. For this reason, logical file SI must first be positioned to the appropriate file before invoking XREF. The NOLO option suppresses the listing of the Assembler source file. Any symbols named LAB1, R1 through R15, ONE, and TWO are not included in the cross-reference. The asterisk (*) in column one is used to terminate XREF command input and begin generating a cross-reference. Any character that is not alphabetic and not a colon could have also been used to terminate command input, including a blank line.

2.3 COMMAND DESCRIPTIONS

XREF commands are described in alphabetical order in the remainder of this chapter.

Bias Line Numbers

The :BIA command is used to automatically adjust line number references by a specified bias. XREF adjusts the line numbers in the cross-reference output listing by adding the number "n" to each source line number. For example, :BIA 5 causes any reference to line 1 to be listed as line 6. This feature is useful to obtain a listing to match a previous listing.

It should be noted that XREF does not reset the bias when it encounters an END directive, so the :BIA command should be used with caution when cross-referencing more than one Assembler program in a single run.

SYNTAX

```
:BIA n
```

where "n" is a bias number.

:CON

Continue if Symbol Table Becomes Full

The :CON command is used to continue cross-referencing even if the symbol table becomes full. If :CON is in effect for the current execution of XREF and the symbol table becomes full, XREF clears the symbol table and starts a new cross-reference. If :CON is not used and the symbol table becomes full before the cross-reference completes, XREF will abort with a reason code of OST.

The $CN option can also be used to accomplish the same function as the :CON command.

SYNTAX

:CON

Terminate Cross-Reference on Double EOF

The :DEO command continues cross-referencing Assembler source programs until XREF encounters a double end-of-file mark. If :DEO is not in effect, the default is for XREF to terminate processing when it encounters an END directive within an Assembler file.

The $DE option can also be used to accomplish the same function as the :DEO command.

To terminate processing when XREF encounters a single end-of-file mark, use the :NOE command or $NE option.

SYNTAX

```
:DEO
```

:EXI

Exit XREF

The :EXI command is used to exit XREF without generating a cross-reference listing.

SYNTAX

```
:EXI
```

No Page Ejects

The :NEJ command is used to generate a double line feed where a page eject would normally occur. :NEJ has no effect when processing reaches the bottom-of-form on the output device.

The $NJ option can be used to accomplish the same function as the :NEJ command.

SYNTAX

:NEJ

:NOE

Terminate on Single EOF

The :NOE command continues cross-referencing Assembler source programs until XREF encounters an end-of-file mark. If :NOE is not in effect, the default is for XREF to terminate processing when it encounters an END directive within the Assembler source file.

The $NE option can also be used to accomplish the same function as the :NOE command.

To terminate processing when XREF encounters a double end-of-file mark, use the :DOE command or $DE option.

SYNTAX

:NOE

Include Numerical and CAN-Code Constants

The :NUM command is used to include numbers and CAN-code constants in the cross-reference listing. The at sign (@) that precedes CAN-code constants and the pound sign (#) that precedes hexadecimal constants are not printed.

The $NM option can also be used to accomplish the same function as the :NUM command.

SYNTAX

:NUM

:OPC

Include Operation Codes

The :OPC command is used to cross-reference symbols that appear in the operation code field of Assembler source statements. The operation code field is the second field of an Assembler source statement. If :OPC is not in effect and symbols that match operation codes are used as labels within the Assembler source program, subsequent references to those symbols will appear on the cross-reference listing.

The :OPC command is useful for cross-referencing macro files, where the operation code field contains a previously defined symbol name.

The $OP option can also be used to accomplish the same function as the :OPC command.

SYNTAX

```
:OPC
```

EXAMPLE

```
        LDI,4   10
          ...
LDI     EQU     $
          ...
        LDI,4   20
```

Assuming that the :OPC command or $OP option is not in effect, the first LDI statement would not be recognized as a symbol by XREF. However, LDI is then used as an actual label on the EQU statement. Therefore, the third and subsequent references to LDI would then be recognized as symbols by XREF and would appear on the cross-reference listing. If the :OPC command or $OP option was in effect for this code, all references to LDI would be recognized as symbols, including all uses of LDI as an instruction in the operation field.

Start Cross-Reference at Specified Line Number

The :STA command is used to start cross-referencing at a specific line number within the Assembler source program. This feature is useful to obtain a partial cross-reference of an Assembler file or macro file. The cross-reference can also be stopped at a particular line number with the :STO command.

SYNTAX

:STA n

where "n" is the line number at which to start cross-referencing.

:STO

Stop Cross-Reference at Specified Line Number

The :STO command is used to stop cross-referencing at a specified line number within an Assembler source file.

When used in conjunction with the :STA command, partial cross-reference listings can be obtained for a range of line numbers within an Assembler source file.

SYNTAX

:STO n

where "n" is the line number at which to stop cross-referencing.

Read Assembler Input from Logical File WRK

The :WRK command is used if Assembler source input is from a random file in Source Editor format on logical file WRK. If :WRK is in effect and the SC option was also specified, XREF ignores the SC option. Any :DEO commands or $DE options are also ignored. XREF reads Assembler source code from logical file WRK until an END directive is encountered.

The $WK option can also be used to accomplish the same function as the :WRK command.

SYNTAX

:WRK

CHAPTER 3
XREF OUTPUT

An Assembler source file and the corresponding cross-reference listing generated by XREF are shown in Figure 3-1 and 3-2. When examining the output illustration, refer to the following sections for explanations of the output format.

3.1 OUTPUT FORMAT

The top line on each page of an XREF listing contains standard information indicating that the output is from the Cross-Reference utility, gives the date and time that the listing was generated, and numbers each page sequentially.

The title specified in the first TTL directive in the Assembler source program appears just before the list of cross-referenced symbols. If the Assembler program does not contain a TTL directive, this line is blank. The symbols and their references are then listed in alphanumeric order, where the sequence of ordering is:

blank space
A - Z
0 - 9
colon (:)
dollar sign ($)
period (.)

The symbol appears on the left-hand side of the page, followed by a three-character description of its usage (Type One indicator), the source line number where that symbol was defined (where it occurred in the label field), and then a list of source line numbers where that symbol was referenced. Some line numbers are then followed by a special character that gives additional information about the use of the symbol at that particular line (Type Two indicator). Line numbers are listed in numerical order. The meanings of the Type One and Two indicators are described in the following sections.

3.1.1 TYPE ONE INDICATORS

A Type One indicator is a three-character string that appears after a symbol name to describe how the symbol was used.

Indicator	Meaning
AOP	The symbol is an Assembly no operation (AOP).
COM	The symbol either defines or references a common area.
CON	The symbol defines a constant.
DEF	The symbol is data which cannot be classified as a constant or a variable. For example, it may be a name used as a parameter to a REX call.

Indicator	Meaning
EQU	The symbol is some type of equate. This can be a compiler-time constant or a label definition to a location other than the current one.
EXT	The symbol represents an external definition or a reference to an external (Assembler EXT directive).
INT	The symbol represents an internal definition (Assembler INT directive).
LAB	The symbol is an actual label (an equate to the current location, e.g., START EQU $).
MAC	The symbol defines the name of a macro.
ORG	The symbol respresents a program origin declaration (Assembler ORG directive).
RES	The symbol is reserving buffer space. If the symbol is later referenced in an Assembler instruction, it will then be marked VAR.
SET	The symbol is setting or resetting its contents to a value.
VAR	The symbol is a variable. The defined data has been referenced as the destination field of an Assembler instruction.
VEC	The symbol is an EQU or DFC symbol through which a branch has been made (vector).
blank spaces	Any nonequated label on a line. This can be as a result of a PGM directive or a label that has not been equated to anything yet (e.g., parameters on a macro call line, a label for a macro call line, or any other nonequated label).

3.1.2 TYPE TWO INDICATORS

A Type Two indicator is a single character that appears after a line number reference. It gives additional information about the use of the symbol at the indicated line.

Indicator	Meaning
blank space	The operation code on this line does not affect the contents of the word denoted by the symbol.
*	Indicates that this line number is the one at which the symbol was initially defined (i.e., where it occurred as a label).

Indicator	Meaning
=	The operation code on this line changes the contents of the word denoted by the symbol.
+	The operation code on this line is a branch to the symbol.

Figure 3-1 below is the Assembler source file used as input to XREF to generate the cross-reference listing shown in Figure 3-2.

```
 1          TTL             READER
 2          PGM             TESTER
 3   *
 4   *  THIS PROGRAM IS DESIGNED TO ILLUSTRATE THE
 5   *  FORMAT OF AN XREF OUTPUT LISTING.
 6   *
 7   NOPE   MAC
 8          AOP
 9          EMP
10   ROPE   MAC
11          AOP
12          EMP
13   AREA   COM             100
14   BREA   COM               1
15   *
16   LAB1   DFC             0000
17   LAB2   DFC             0001
18   LAB3   DFC             0003
19   LAB5   DFC             0005
20   FRED   DFC             0016
21   ZERO   EQU                0
22   TEN    EQU               10
23   HUN    EQU              100
24   VAN    SET               TEN
25   *
26   BUFF   RES                5
27   *
28   BGF    ORG             1000
29   START  EQU             $
30          LDM,2           FRED
31          LDM*5           TEN
32          STM,2           LAB3
33   LAB4   LDI,3           TEN
34   ALPHA  CEQ             AREA+1
35   HOPE   AOP
36   *
37   SUBR   EXT             SKA
38   SUB2   INT             LAB5
39          END             START
```

Figure 3-1. Assembler Source Input to XREF

```
      TTL                READER
ALPHA    COM    34*
AOP      OPC     8        11       35
AREA     COM    13*       34
BGF      ORG    28*
BREA     COM    14*
BUFF     RES    26*
CEQ      OPC    34
COM      OPC    13        14
DFC      OPC    16        17       18       19       20
EMP      OPC     9        12
END      OPC    39
EQU      OPC    21        22       23       29
EXT      OPC    37
FRED     CON    20*       30
HOPE     AOP    35*
HUN      EQU    23*
INT      OPC    38
LAB1     DEF    16*
LAB2     DEF    17*
LAB3     VAR    18*       32=
LAB4            33*
LAB5     DEF    19*       38
LDI      OPC    33
LDM      OPC    30        31
MAC      OPC     7        10
NOPE     MAC     7*
ORG      OPC    28
PGM      OPC     2
RES      OPC    26
ROPE     MAC    10*
SET      OPC    24
SKA             37
START    LAB    29*       39
STM      OPC    32=
SUBR     EXT    37*
SUB2     INT    38*
TEN      CON    22*       24       31       33
TESTER           2
VAN      SET    24*
ZERO     EQU    21*
0                21
0000             16
0001             17
0003             18
0005             19
0016             20
1                14        34
10               22
100              13        23
1000             28
2                30        32=
3                33
5                26
```

Figure 3-2. XREF Output Listing

3.2 DEVICE LIMITATIONS FOR PRINTED OUTPUT

Depending on the type of output device to which XREF output listings are directed, certain limitations may be applicable as described below.

If XREF's printed output is directed to a device that does not provide an end-of-form indication, the listing will not be divided into pages. Instead, each report will be delineated by a new heading.

If XREF's printed output is directed to a device that cannot display records as long as those in the Assembler source input, the records will be truncated to that device's physical record size and some information will be lost.

If XREF's printed output is directed to the same device from which it is reading commands and the device supports both input and output operations, the commands will appear twice on the listing, once as they are entered and once as they are listed.

If XREF's printed output is directed to a slow device, it may be desirable to suppress the Assembler source listing by using the NOLO option.

3.3 ERROR MESSAGES

The following error messages may be generated when an error occurs during XREF processing.

COMMAND ERROR
: An invalid command was specified and logical file CI or AI is assigned to a terminal device. If logical file CI or AI is not assigned to a terminal device and a command is specified incorrectly, XREF will abort with a reason code of ILD.

'END' LINE NOT PRESENT
: An end-of-file was encountered before an Assembler END directive.

INVALID RANDOM FORMAT ON WRK
: The $WK option was set or the :WRK command was entered and the format of the Assembler input on logical file WRK is not valid random format.

INDEX

accessing XREF, 1-4

biasing line numbers, 2-3

cataloging XREF, 1-5
command input, 1-3, 2-1
command summary, 1-2
constants, 2-9
cross-referencing insert files, 1-2
cross-referencing operation codes, 1-5, 2-10

error messages, 3-5
exiting XREF, 2-6

file marks, 2-5, 2-8

indicators on listed output, 3-1
input
  Assembler programs, 1-3, 1-4, 2-13
  XREF commands, 1-3, 2-1
insert files, 1-2

Job Control to execute XREF, 1-5

logical files used by XREF, 1-3, 2-1, 2-13

operation codes, 1-1, 1-5, 2-10
options, 1-2, 1-3, 1-4
output device limitations, 3-5
output example, 3-4
output format, 3-1

partial cross-references, 2-11, 2-12
printed output limitations, 3-5
procedure to execute XREF, 1-5

symbol suppression, 2-1
symbol table, 1-6, 2-4
symbols, 1-1

titles on output listings, 3-1
type one indicators, 3-1, 3-2, 3-4
type two indicators, 3-2, 3-3, 3-4

workfile, 1-3, 1-4, 2-13

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 3624 FT. LAUDERDALE, FL 33309

POSTAGE WILL BE PAID BY ADDRESSEE

MODULAR COMPUTER SYSTEMS
1650 W. McNAB ROAD
P.O. BOX 6099
FT. LAUDERDALE, FLORIDA 33310

Attention: TECHNICAL PUBLICATIONS, M.S. #85

Fold

MODCOMP®

Please comment on the publication's completeness, accuracy, and readability. We also appreciate any general suggestions you may have to improve this publication.

If you found errors in this publication, please specify the page number or include a copy of the page with your remarks.

Your comments will be promptly investigated and appropriate action will be taken. If you require a written answer, please check the box and include your address below.

☐

Comments: _____

Manual Title _____

Manual Order Number _____ Issue Date _____

Name _____ Position _____

Company _____

Address _____

_____ Telephone ( ) _____

Corporate Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 974-1380, TWX: 310-372-7837

International Headquarters:
MODULAR COMPUTER SERVICES, Inc., The Business Centre, Molly Millars Lane, Wokingham, Berkshire, RG11 2JQ, UK, Tel: 0734-786808, TLX: 851849149

Latin American Sales Headquarters:
MODULAR COMPUTER SYSTEMS, Inc., 1650 West McNab Road, P.O. Box 6099, Ft. Lauderdale, FL 33310, Tel: (305) 975-6562, TLX: 3727852

Canadian Headquarters:
MODCOMP Canada, Ltd., 400 Matheson Blvd. East, Unit 24, Mississauga, Ontario, Canada L4Z 1N8, Tel: (416) 890-0666, TELEX: 06-961279

SALES & SERVICE LOCATIONS THROUGHOUT THE WORLD

"The technical contents of this document, while accurate as of the date of publication, are subject to change without notice."

Printed in the U.S.A.